

CS 4476: Computer Vision, Fall 2020

PS0

Author: Saumya Jain

Date: Wednesday, August 26th, 2020

## Problem 1

(a) `x = np.random.permutation(1000)`

This snippet created an array using the first 1000 integers (0-999). These integers are shuffled randomly inside the array. Here, the `random.permutation` function uses `arange` to make an array of 1000 elements (the parameter passed in) and then shuffles it to permute the array.

(b) `a = np.array([[1,2,3],[4,5,6],[7,8,9]])`  
`b = a[2,:]`

In the first line, an array matrix is created with the values in the parameter and assigned to variable **a**. In the second line, we use slicing to pull out the subarray consisting of the third row. **b** is assigned the value `[7,8,9]`

(c) `a = np.array([[1,2,3],[4,5,6],[7,8,9]])`  
`b = a.reshape(-1)`

In the first line, an array matrix is created with the values in the parameter and assigned to variable **a**. In the second line, we use the `reshape` function to change the shape of the array matrix based on the parameter, while causing no changes to the array matrix's contents. Here, the parameter is `-1` which converts our array matrix **a** to a single array with all the elements. Now, variable **b** is a single array: `[1,2,3,4,5,6,7,8,9]`

(d) `x = np.random.randn(5,1)`  
`y = x[x>0]`

In the first line, `np.random.randn` is used to generate an array of random floats sampled from the “standard normal” distribution. This distribution has mean = 0 and variance = 1. In the second line, we are going through our created array **x** and generating an array of only values greater than 0. This is an array assigned to **y**.

(e) `x = np.zeros(10)+0.5`  
`y = 0.5*np.ones(len(x))`  
`z = x + y`

In the first line, we generate a 10 element array where every element is 0. By adding 0.5 to every element, we end up with **x** being a 10 element array where every element is 0.5. In the second line, we create an array of length of **x** (which is 10) and populate it with all 1's. Then, every element is multiplied by 0.5 to result in **y** being a 10 element array where every element is 0.5. In the third line, we add arrays **x** and **y**. Since these are both 10 element arrays with every element as 0.5, we end up with **z** being a 10 element array with every element as 1. So **z** = `[1,1,1,1,1,1,1,1,1,1]`

(f) `a = np.arange(1,100)`  
`b = a[::-1]`

In the first line, we generate a 99 element array of integers between 1 (inclusive) and 100 (exclusive). In the second line, we are reversing the entire array `a` and assigning that to `b`. So `b` is the array: `[99,98,97...3,2,1]`

## Problem 2

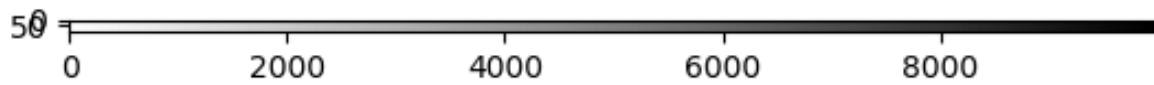
(a) `def random_dice(N):`  
    `a = np.random.rand(N)`  
    `b = (a * 5) + 1`  
    `outcomes = np rint(b).astype(np.uint8)`  
    `return outcomes`

(b) `def reshape_vector(y):`  
    `return np.reshape(y, (3,2))`

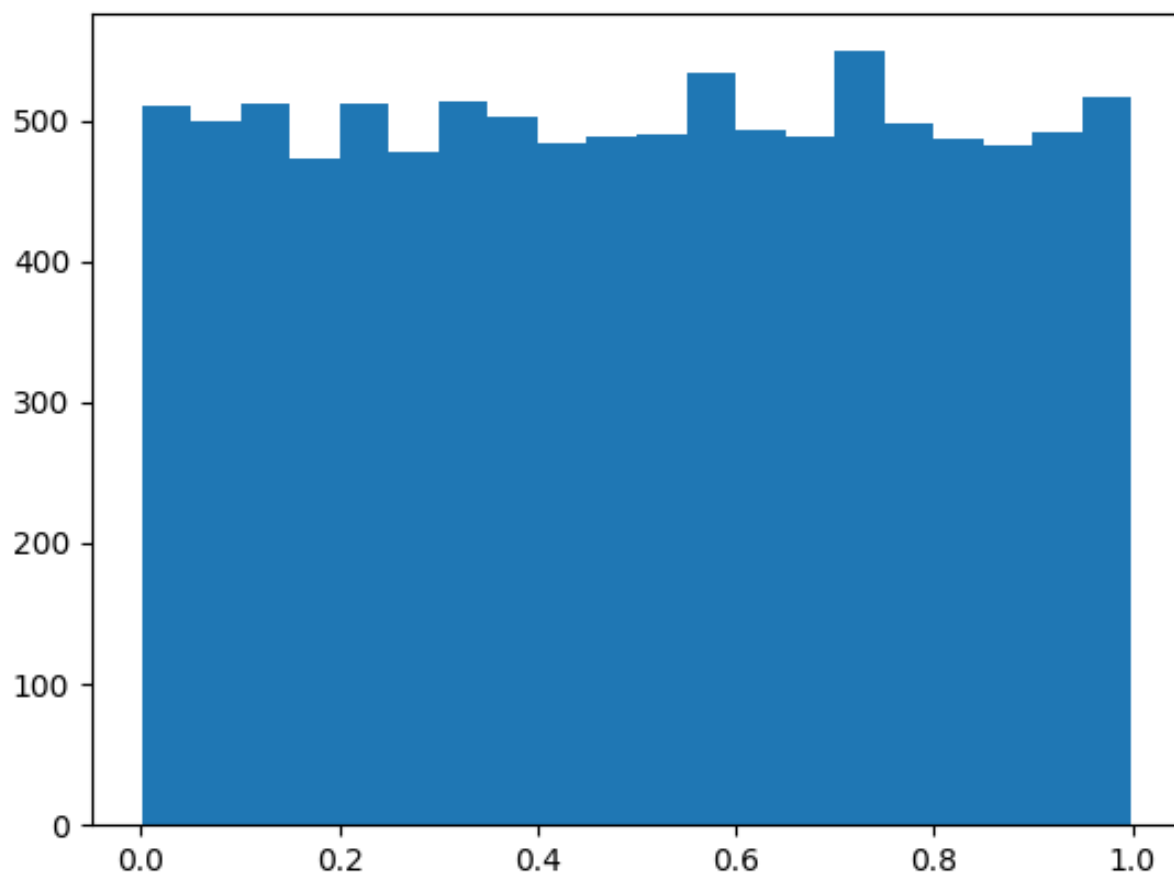
(c) `def max_value(z):`  
    `max_val = np.where(z == np.max(z))`  
    `return (max_val[0][0], max_val[1][0])`

(d) `def count_ones(v):`  
    `return np.count_nonzero(v == 1)`

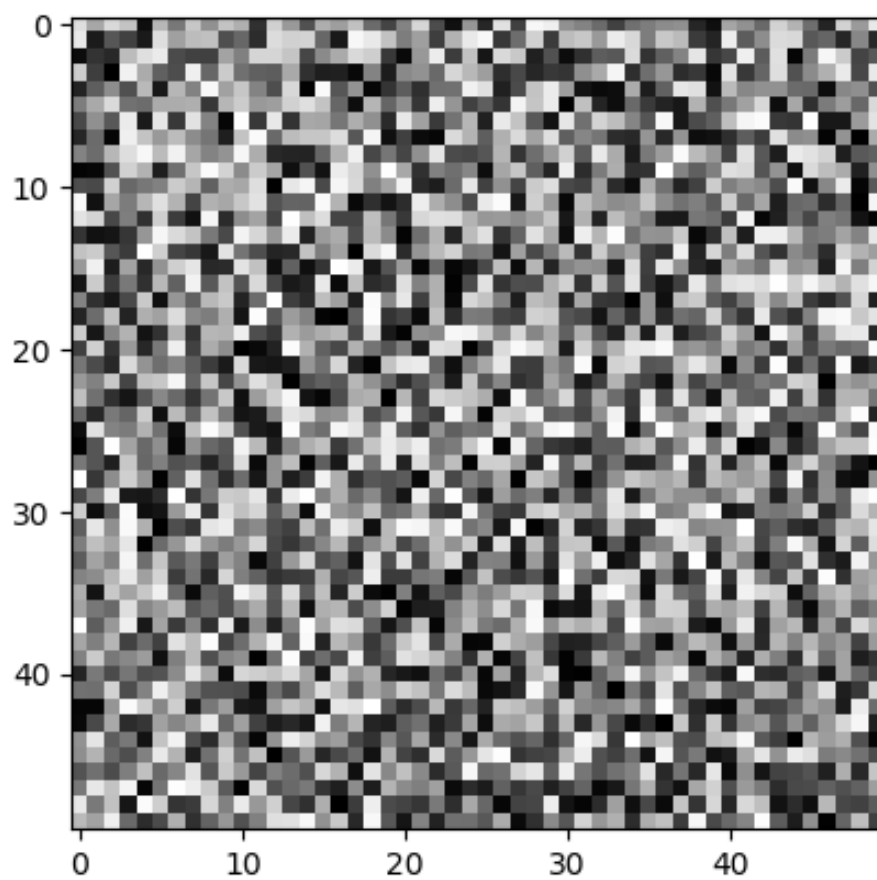
## Problem 3



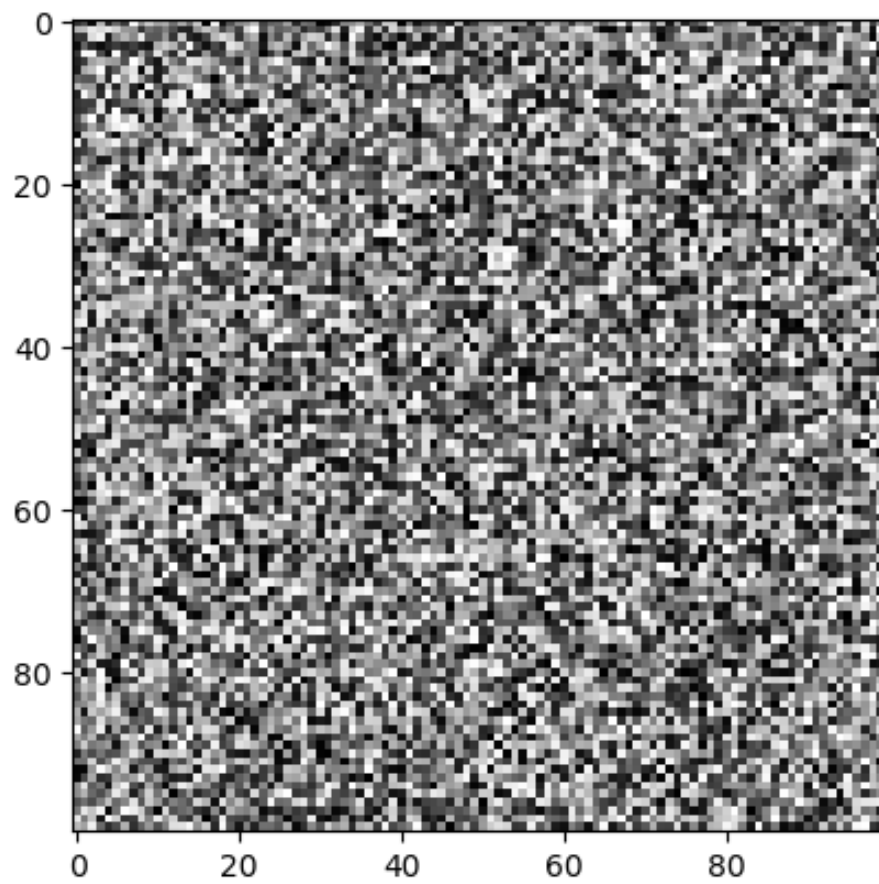
Sorted Intensities - Question 3(a)



Intensity Histogram - Question 3(b)

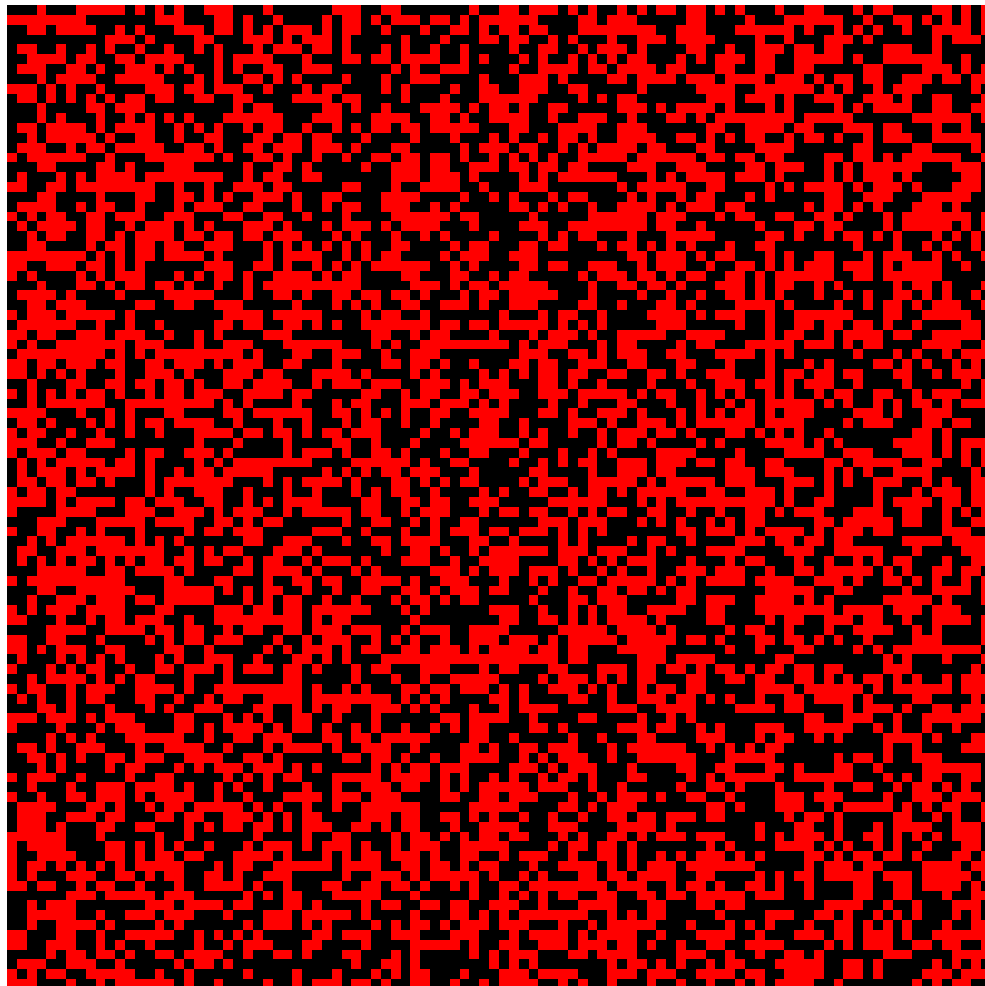


X - Bottom Left Quadrant - Question 3(c)



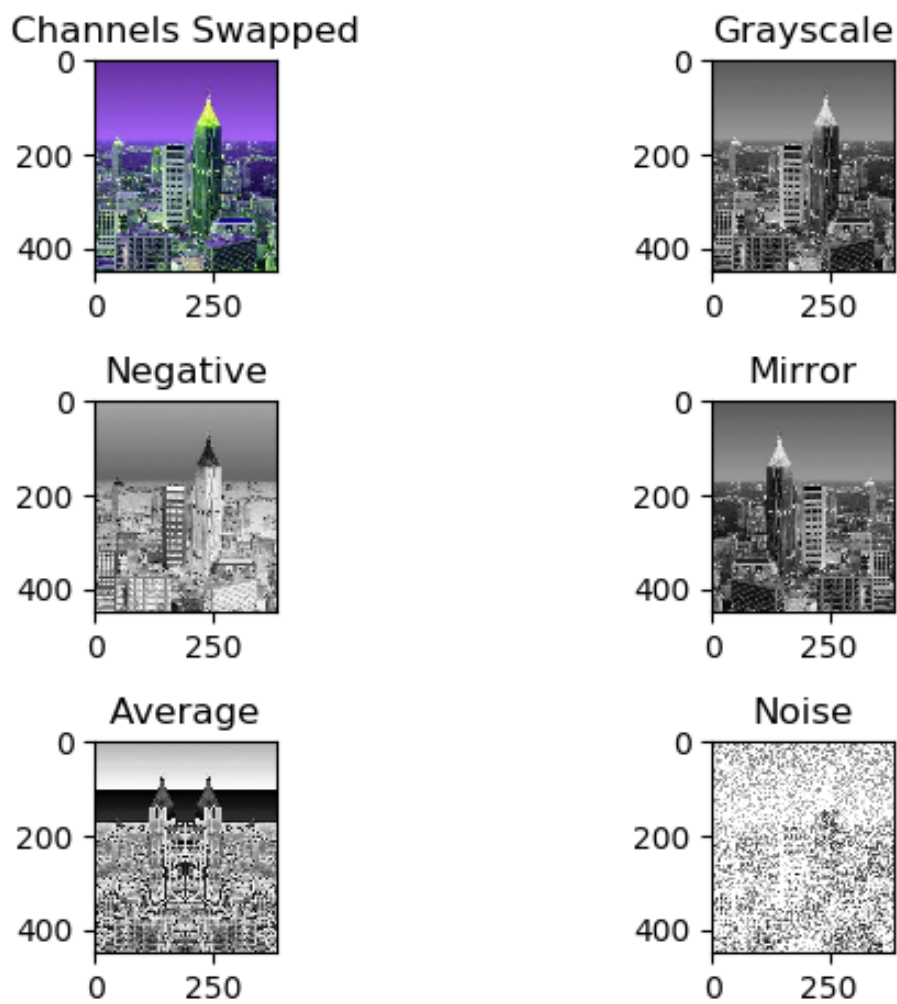
Y- Mean Subtracted - Question 3(d)





Color Image - Question 3(e)

## Problem 4



Subplot - Question 4)